UiO : University of Oslo

Structured Probabilistic Modelling for Dialogue Management

Pierre Lison
Language Technology Group
University of Oslo

ESSLLI 2015, Barcelona

Outline for this talk

- The dialogue management task
- A hybrid logical/probabilistic approach
  - Probabilistic rules
  - Parameter estimation
  - Experiments
- Demonstration of the OpenDial toolkit

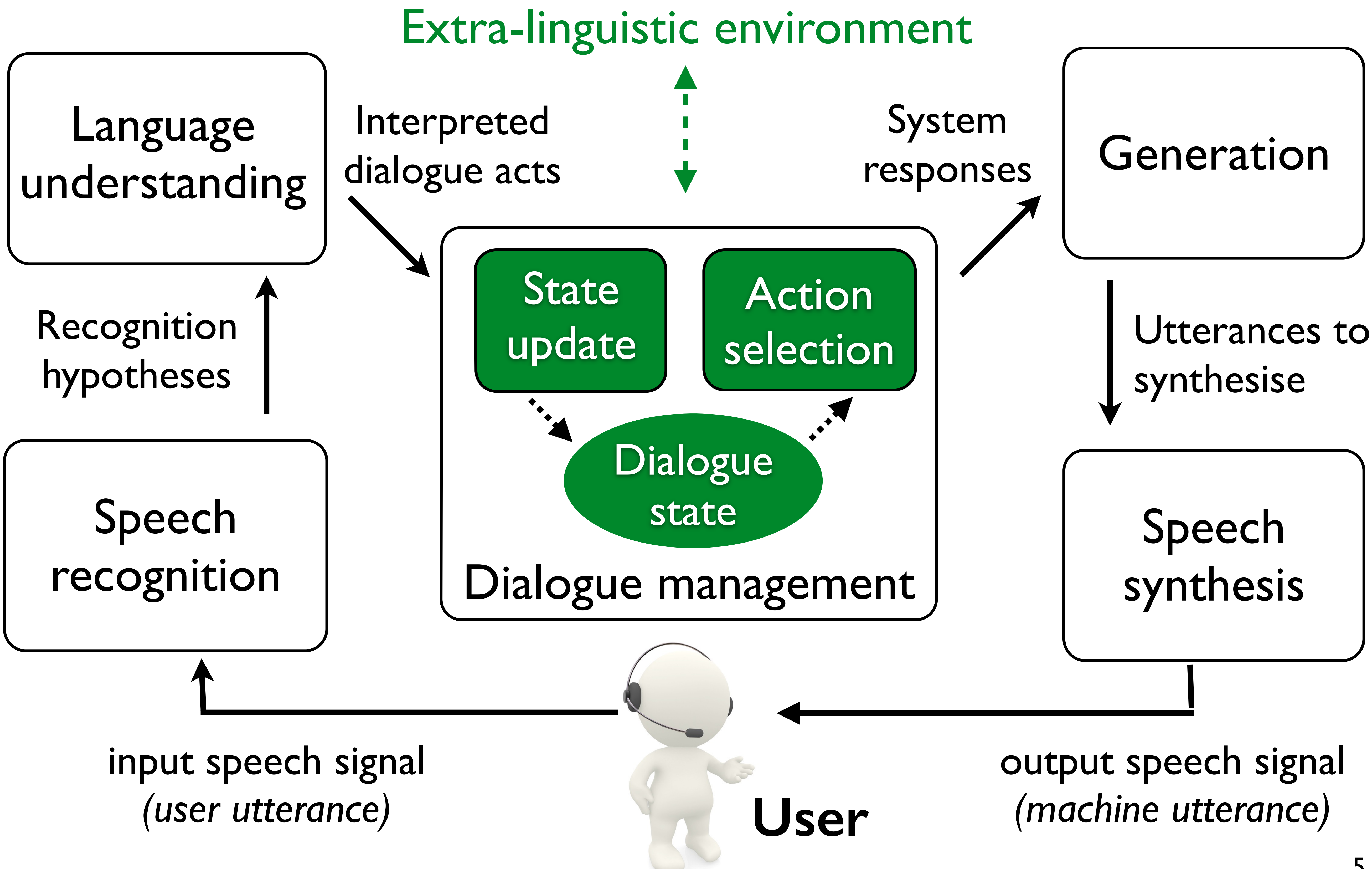
Outline for this talk

- **The dialogue management task**
- A hybrid logical/probabilistic approach
  - Probabilistic rules
  - Parameter estimation
  - Experiments
- Demonstration of the OpenDial toolkit

What is dialogue management?

- A component in (spoken) dialogue systems
- In charge of "managing" the interaction
  - Maintain a representation of the current *state* of the dialogue
  - Select the next system *actions* based on this state
  - *Predict* how the interaction is going to unfold
- Difficult problem!
  - Dialogue is *complex* (many contextual factors to capture)
  - Dialogue is *uncertain* (ambiguities, unexpected events, etc.)

Typical dialogue architecture



Existing DM techniques

	Logical approaches	**Statistical approaches**
+	Fine-grained control of conversation	Robust, data-driven models of dialogue
-	Limited account for uncertainties	Need large quantities of training data

A new **hybrid** modelling framework based on *probabilistic rules*

Outline for this talk

- What is dialogue management?
- **A hybrid logical/probabilistic approach**
  - **Probabilistic rules**
  - Parameter estimation
  - Experiments
- Demonstration of the OpenDial toolkit

The key idea

- We start with the core ideas behind probabilistic dialogue modelling:
  - Dialogue state represented as a **Bayesian Network**
  - Each variable captures a *relevant aspect of the interaction* (dialogue history, user intentions, external context, etc.)
  - The dialogue state is regularly updated with *new observations* (spoken inputs, new events), according to domain-specific probabilistic models
  - ... and used to determine the *next actions* to execute, according to domain-specific utility models

The key idea

But:

- instead of expressing the domain models using traditional formats (e.g. probability tables)...
- ... we adopt a high-level representation based on **probabilistic rules.**
- The probabilistic rules provide an *abstraction layer* on top of probabilistic (graphical) models

Less parameters to estimate (=easier to learn from small amounts of data)

Can express expert knowledge in human-readable form

Rule structure

- Basic skeleton: *if... then...else* construction:

$\forall$ x, **if** (condition$_1$ holds) **then**
...
else if (condition$_2$ holds) **then**
...
else
...

- Mapping between *conditions* and (probabilistic) *effects*
- Can use logical operators and universal quantifiers
- Two types of rules: *probability* and *utility* rules

Two types of rules

	Probability rules	**Utility rules**
What they encode:	*Conditional probability distributions* between state variables	*Utility functions* for system actions given state variables
General structure:	**if** (condition_1) **then** $P(\text{effect}_1) = \theta_1$, $P(\text{effect}_2) = \theta_2$, ... **else if** ($\text{condition}_2$) **then** $P(\text{effect}_3) = \theta_3$, ...	**if** (condition_1) **then** $U(\text{action}_1) = \theta_1$, $U(\text{action}_2) = \theta_2$, ... **else if** ($\text{condition}_2$) **then** $U(\text{action}_3) = \theta_3$,... ...

Examples of probabilistic rules

$\forall$ x,
if (*last-user-act* = *x* $\wedge$ *system-action* = AskRepeat) **then**
P(*next-user-act* = *x*) = 0.9

"If the system asks the user to repeat his last dialogue act *x*, the user is predicted to comply and repeat *x* with probability 0.9"

$\forall$ x,
if (*last-user-act*=Request(*x*) $\wedge$ x $\in$ *perceived-objects*) **then**
U(*system-action*=PickUp(x)) = +5

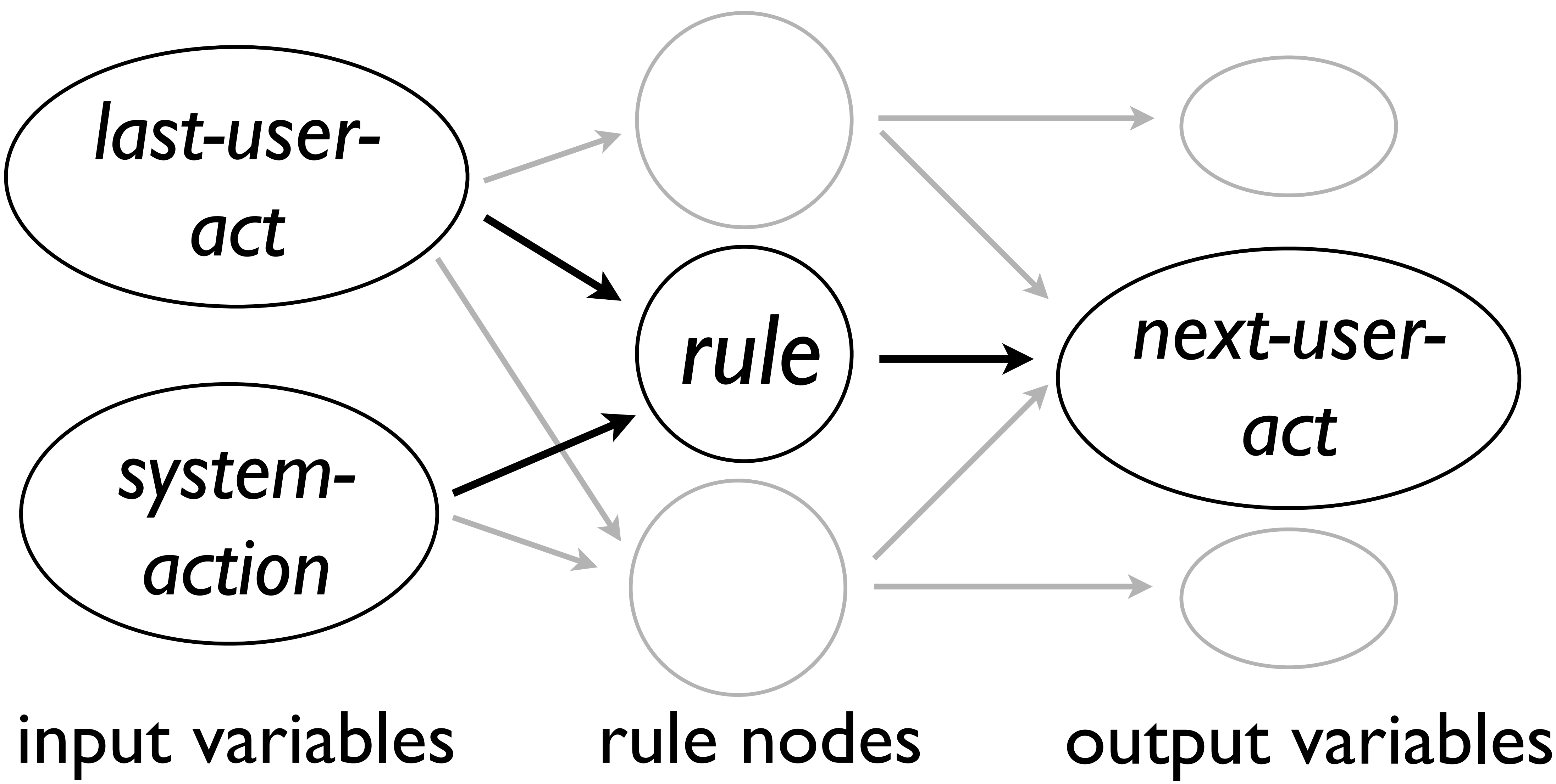
"If the user asks the system to pick up a given object *x* and *x* is perceived by the system, then the utility of picking up *x* is 5"

Rule instantiation

- At runtime, the rules are "executed" by instantiating them in the dialogue state:
  - The rules can be seen as "high-level templates" for the generation of a classical probabilistic model
  - Inference (for state update and action selection) is then performed on this grounded representation
- The use of logical abstractions allows us to capture complex relations between variables in a compact, human-readable form

Instantiation of probability rules

$\forall$ x,
if (*last-user-act* = *x* $\wedge$ *system-action* = AskRepeat) **then**
P(*next-user-act* = *x*) = 0.9

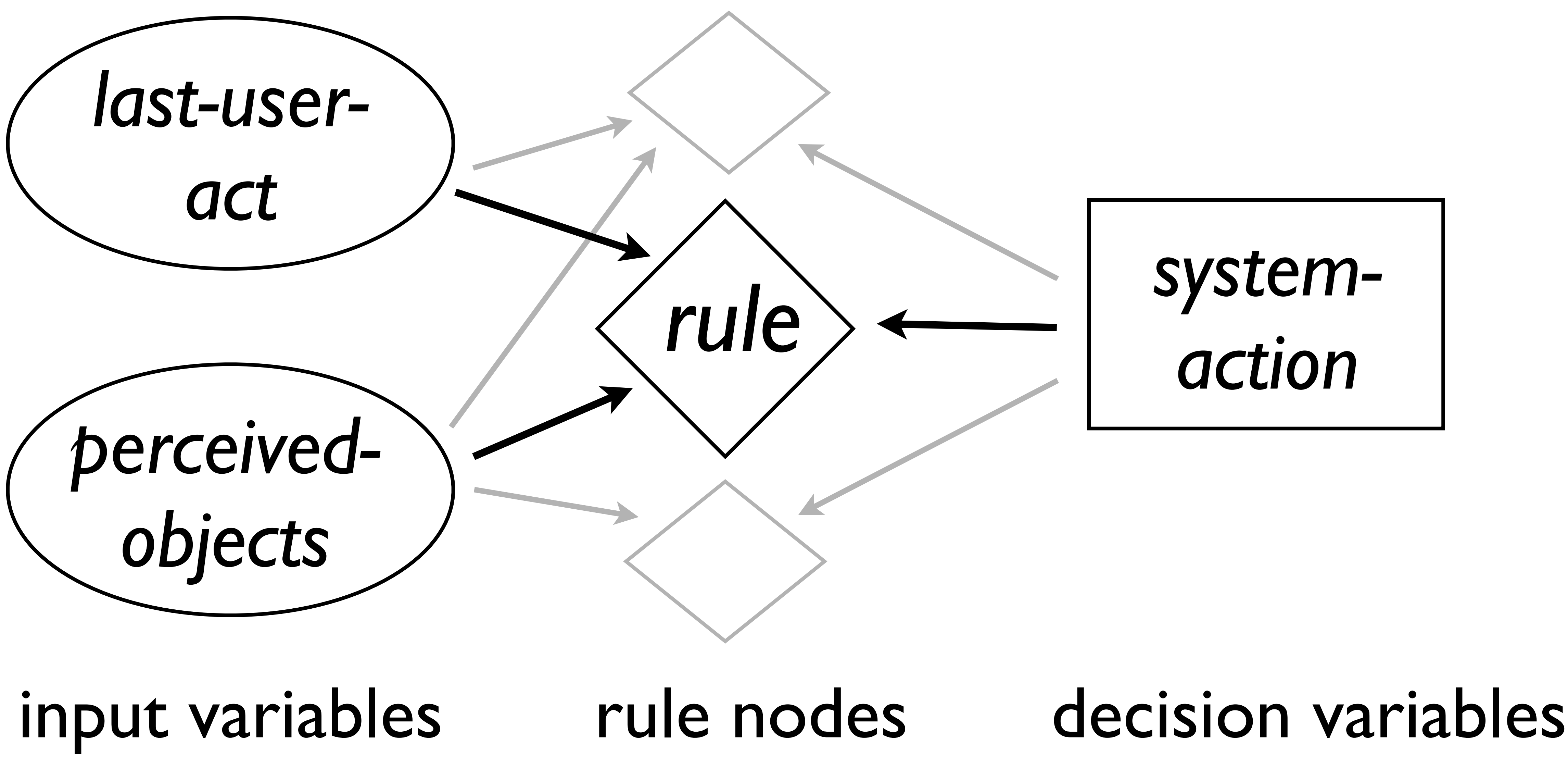


Instantiation of utility rules

∀ x,

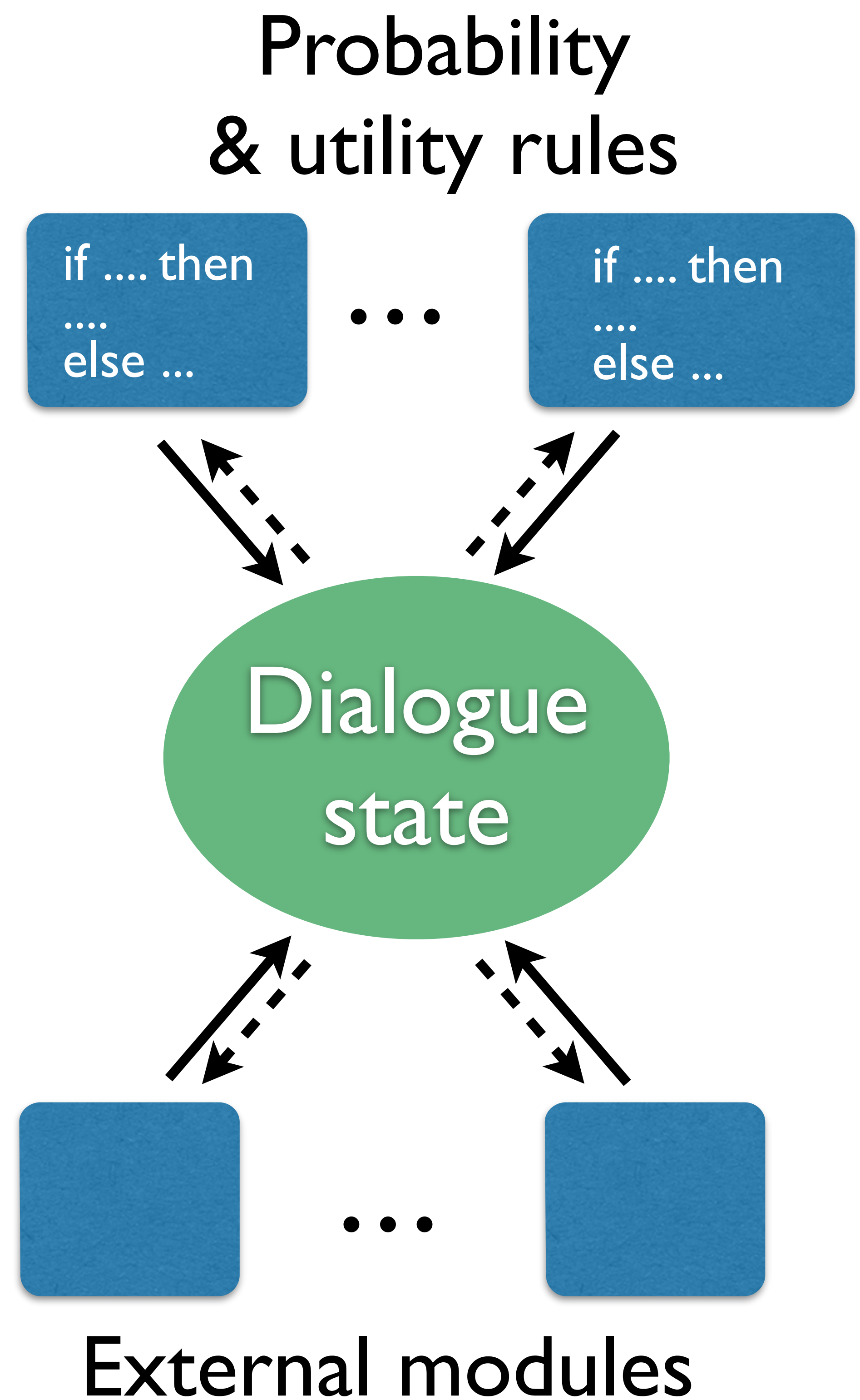
if (*last-user-act*=Request(x) ∧ x ∈ *perceived-objects*) **then**

U(*system-action*=PickUp(x)) = +5



Processing workflow

- *Information state architecture*, with the dialogue state encoded as a Bayesian Network
- External modules (e.g. ASR, vision) add new observations
- Probability rules employed to update the dialogue state (following the new observations)
- Utility rules employed to determine the system actions
- Implementation: OpenDial toolkit [http://www.opendial-toolkit.net]



Domain representation

- Dialogue domains are represented in OpenDial in an XML format containing:
  - The initial dialogue state for the interaction
  - A collection of *domain models* (see below)
  - A collection of external modules & their configuration
- The domain models are simply a collection of (probability or utility) rules that are triggered by a common update event
  - Each model represents a distinct "processing step"

Outline for this talk

- What is dialogue management?
- **A hybrid logical/probabilistic approach**
  - Probabilistic rules
  - **Parameter estimation**
  - Experiments
- Demonstration of the OpenDial toolkit

Parameter estimation

- Probabilistic rules may include parameters (unknown probabilities or utilities)
- **Bayesian learning** approach:
  - Start with initial prior over possible parameter values
  - Refine the distribution given the observed data $\mathcal{D}$

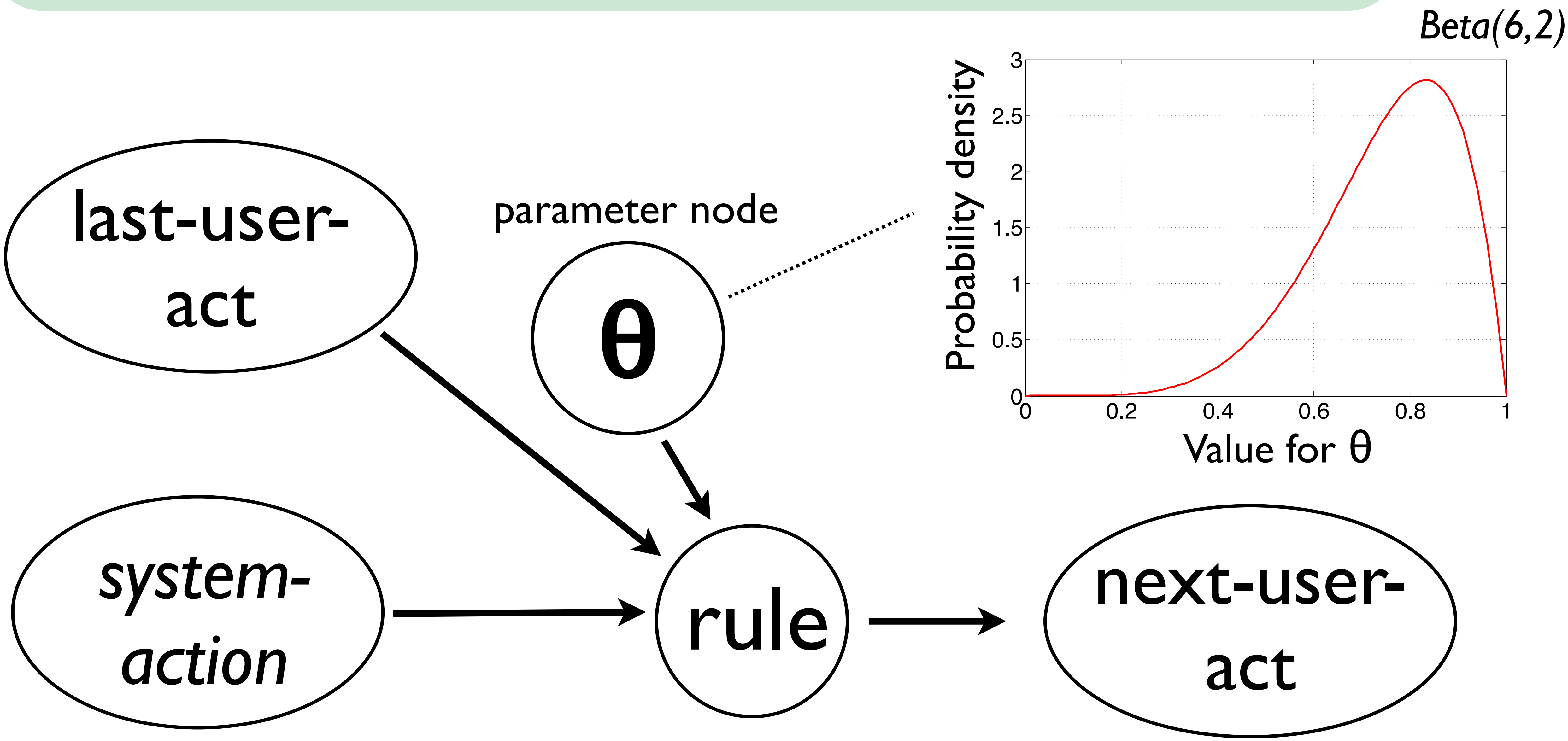
$$\underbrace{P(\boldsymbol{\theta} \mid \mathcal{D})}_{\text{Posterior distribution}} = \underbrace{\eta}_{\text{Normalisation factor}} \underbrace{P(\mathcal{D} ; \boldsymbol{\theta})}_{\text{Likelihood of the data}} \underbrace{P(\boldsymbol{\theta})}_{\text{Prior distribution}}$$

Parameter estimation

∀ x,

if (*last-user-act* = *x* ∧ *system-action* = AskRepeat) **then**

P(*next-user-act* = *x*) = θ



Learning paradigms

- Different types of training data:
  - *Supervised learning*: Wizard-of-Oz interactions

Goal: find the parameter values that best "imitate" the Wizard's conversational behaviour

  - *Reinforcement learning*: real or simulated interactions

Goal: find the parameter values that provide the best fit for the collected observations

[P. Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech & Language*, 2015]

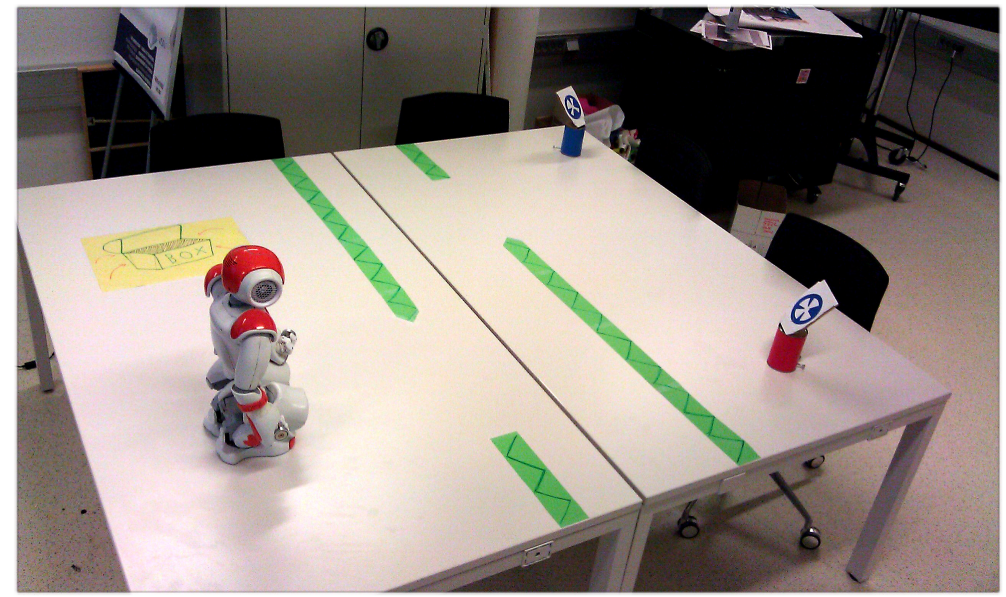
[P. Lison. Model-based Bayesian Reinforcement Learning for Dialogue Management (Interspeech 2013)]

Outline for this talk

- What is dialogue management?
- **A hybrid logical/probabilistic approach**
  - Probabilistic rules
  - Parameter estimation
  - **Experiments**
- Demonstration of the OpenDial toolkit

User evaluation

- **Task**: instruct the robot to move across the table, pick one cylinder and release it on the landmark



- Comparison of three modelling approaches:
  1. A handcrafted finite-state automaton
  2. A factored statistical model
  3. A model structured with probabilistic rules

Experimental procedure

- **Step 1:** collect Wizard-of-Oz interaction data
- **Step 2:** Estimate the internal parameters for the 3 models with the collected data
- **Step 3:** Conduct user trials for the 3 approaches
- **Step 4:** Compare them on dialogue quality metrics

Dialogue domain:

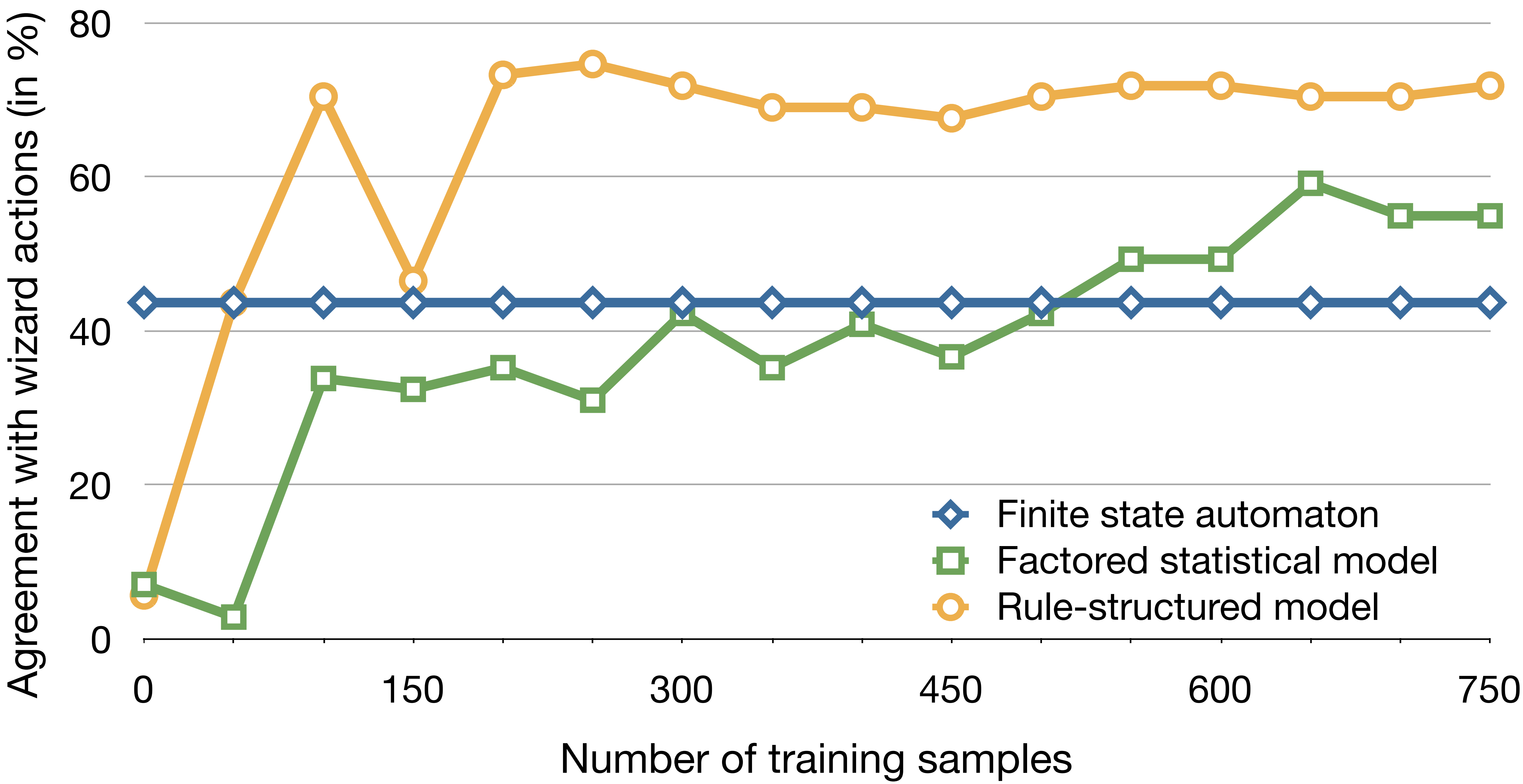
- 26 user actions
- 41 system actions
- State size: 35×10^6 (10 variables)

Parameter estimation:

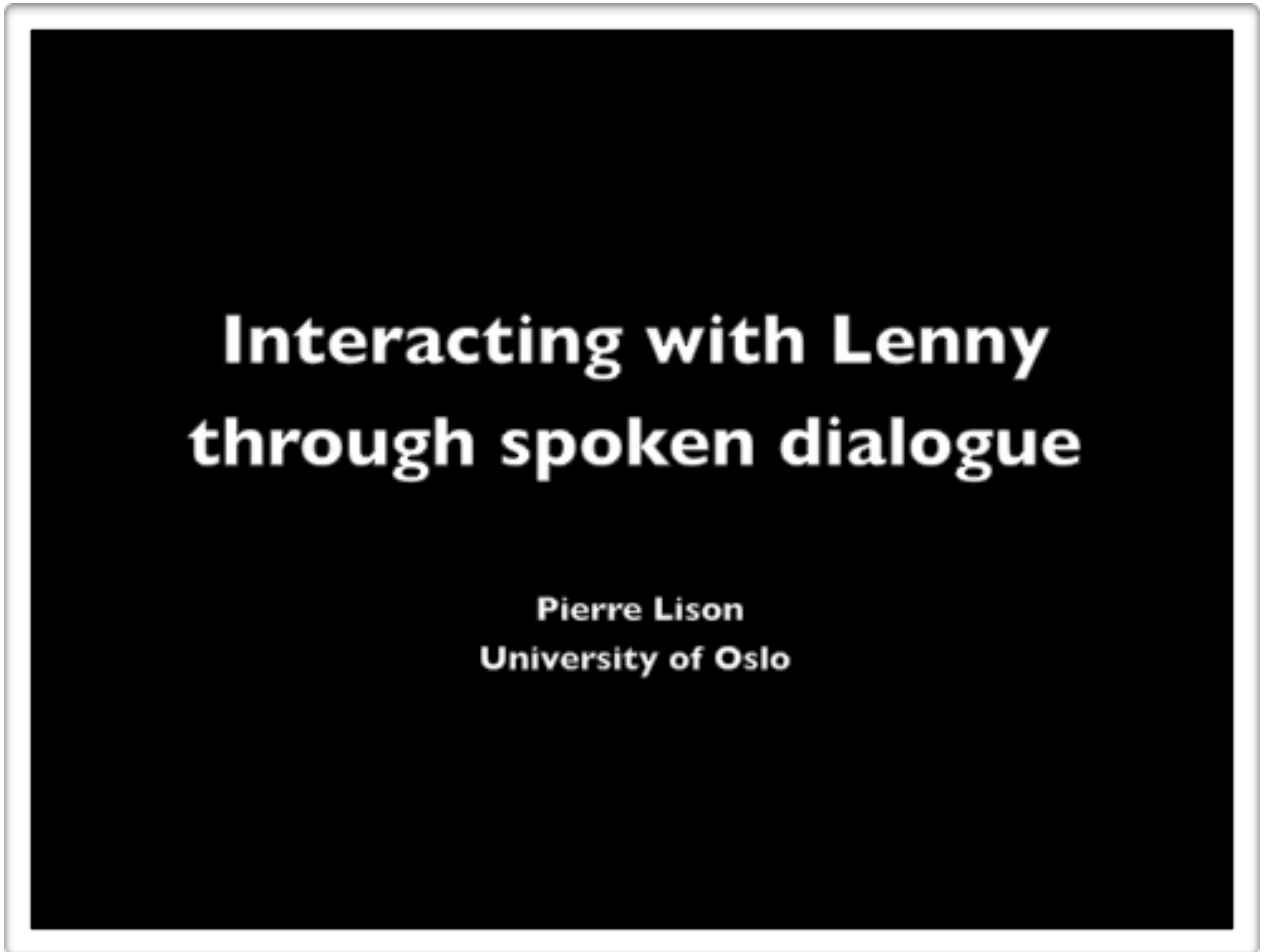
- 10 recorded WoZ interactions
- 3 parameters in handcrafted automaton (thresholds)
- 433 parameters in factored statistical model
- 28 parameters in model encoded with probabilistic rules

Learning curve

Training: 9 Wizard-of-Oz interactions (770 system turns)
Testing: 1 Wizard-of-Oz interaction (71 system turns)



User trials



- 37 participants (16 M / 21 F)
- Average age : 30.6
- Average duration: 5:06 mins
- All captured on videos

User trials

- Each participant in the trial repeated the task **three** times
  - One interaction for each modelling approach (in randomised order)
- Evaluation metrics:
  - **Objective metrics**: list of 9 measures extracted from the interaction logs
  - **Subjective metrics** : survey of 6 questions filled by the participants after each interaction

Empirical results

	Metrics	Finite-state automaton	Factored statistical model	Rule-structured model
Objective	Average number of repetition requests	18.68	12.24	**0***
	Average number of confirmation requests	9.16	10.32	**5.78***
	Average number of repeated instructions	3.73	7.97	**2.78**
	Average number of user rejections	**2.16**	2.59	2.59
	Average number of physical movements	**26.68**	29.89	27.08
	Average number of turns between moves	3.63	3.1	**2.54***
	Average number of user turns	78.95	77.3	**69.14**
	Average number of system turns	57.27	54.59	**35.11***
	Average duration (in minutes)	6:18	7:13	**5:24***
Subjective	*"Did you feel that…*			
	… the robot correctly understood what you said?"	3.32	2.92	**3.68**
	… the robot reacted appropriately to your instructions?"	3.70	3.32	**3.86**
	… the robot asked you to repeat/confirm your instructions?"	2.16	2.19	**3.3***
	… the robot sometimes ignored when you were speaking?"	3.24	2.76	**3.43**
	… the robot thought you were talking when you were not?"	3.43	3.14	**4.41***
	… the interaction flowed in a pleasant and natural manner?"	2.97	2.46	**3.32**

Scale from 1 (worse) to 5 (best)

Outline for this talk

- The dialogue management task
- A hybrid logical/probabilistic approach
  - Probabilistic rules
  - Parameter estimation
  - Experiments
- Demonstration of the **OpenDial** toolkit

Conclusion

- Development of a new modelling framework for dialogue management, based on **probabilistic rules**
  - *Hybrid* approach at the crossroads between logical and statistical methods
  - Rule parameters can be learned from data
- Experimental studies demonstrate the benefits of the approach
- Concrete implementation in the OpenDial software toolkit